

## REMARKS

The Advisory Action dated June 27, 2005 maintained the rejection of the claims. Applicant filed a notice of appeal but upon reflection filed a RCE and this amendment because it is believed that there are fundamental differences between applicant's invention and the references that the examiner has relied upon. Accordingly, the claims have been amended to emphasize those differences and to place the application in condition for immediate allowance.

A fundamental difference between applicant's invention and Jagannathan is the feature of applicant's automatic thread migration compared to the manual migration process which Jagannathan discloses. Jagannathan requires a programmer to write code to facilitate the migration of agents or subparts thereof. This is in sharp contrast to applicant's invention wherein automatic migration occurs, with the when and where of the migration is based upon monitoring the environments in which each of the agents executes. Jagannathan simply does not operate or suggest operating in this manner.

Jagannathan requires the programmer of the system to explicitly write code of the migration methods within each of the agent, object and task classes, respectively. This is specifically disclosed with regard to figure 6 at column 13, lines 23-50:

First, a plurality of object-oriented classes, including an object class, a base class, an agent class and a task class, are defined in a step 80. Next, an object migrate method is defined in the object class in a step 81. When called, the object migrate method migrates a selected object instance to a location specified with the base class (i.e. a base instance). In a step 82, a task migrate method is defined in the task class which, when called, migrates a selected task instance to a location specified with the base class. Similarly, in a step 83, an agent migrate method is defined in the agent class which, when called, migrates a selected agent process to a location specified with the base class.

After the migrate methods have been defined, an agent process is instantiated according to the agent class in a step 84. The agent process may include or encapsulate task instances instantiated according to the task class and object instances instantiated according to the object class. The agent instantiated in step 84 is distributed among the plurality of computer machines of the network, so the task instances and object instances, like the agent process itself, may similarly be distributed among the computer machines of the network. In a step 85, the object migrate method, task migrate method and agent migrate method are performed within the agent process. It should be noted that the methods performed during step 85 need not be performed in any particular order, and each may be performed multiple times, if desired. Moreover, only some of the migrate methods may be defined and performed, if desired.

From the above, it is clear that not only does a programmer have to “hand code” the migration of the agent, but the agent itself has to internally initiate that migration process.

The present invention specifically relates to a process for providing automatic thread migration whereby there is no need for a programmer of the system to do any additional coding to facilitate the migration of any agents. Components constantly monitor the environment for indications that would trigger an automatic migration, such as for example, memory utilization or network latency. In addition, strong thread migration can still take place with applicant’s invention even if the agent itself or the machine upon which it is executing is destroyed or rendered inoperative or the network connection to the machine is lost.

Turning to the claims and particularly amended claim 1, A method for providing parallel execution of computing tasks in a heterogeneous computing environment having a plurality of computing hosts, that comprises, *inter alia*, the step of monitoring the computing hosts for detecting an indication that migration needs to be initiated, determining available computing hosts in the heterogeneous computing environment, and automatically transferring said mobile agents to determine available

computing hosts responsive to a detected indication that migration needs to be initiated. These steps are simply not performed, taught or remotely suggested by Jagannathan applied singularly or in combination with any of the other prior art of record. The transfer of mobile agents by Jagannathan occurs only as a result of a manual process as described in the text set out above in connection with Figure 6 of the Jagannathan patent. Reconsideration and allowance of this claim is respectfully requested.

Moreover, the detected indication of claim 1 is further defined as comprising an indication of network latency in claim 2, or an indication of hostile attack as set forth in claim 16, or an indication of hacking as set forth in claim 17, or an indication of network failure as set forth in claim 18 or an indication of computer hardware failure as indicated in claim 19.

With regard to amended claim 11, it is directed to a method for providing parallel computing using mobile agents, and comprises, *inter alia*, the step of monitoring the computing host for detecting an indication that migration needs to be initiated as well as the step of transferring the one or more mobile agents to one or more available computing hosts responsive to a detected indication that migration needs to be initiated, *inter alia*. For the reasons that have been described with regard to claim 1, this claim is also believed to be in condition for immediate allowance and such action is respectfully requested.

With regard to amended claim 14 which is directed to a method for migrating a software application running in a virtual machine from a primary host to a secondary host, it comprises, *inter alia*, the step of detecting an indication to migrate the application, wherein said indication comprises one of the group consisting of network latency, hostile attack, hacking, network failure or computer hardware failure, and the step of: in response to a detected indication, automatically migrating the application in its entirety from said first computing host to a third computing host without modifying the virtual machine at said third computing host by reconstructing each of the plurality of mobile agents at said third computer host using said stack trace and state information.

Since neither Jagannathan or any of the other references of record teach or suggest these steps, reconsideration and allowance of this claim is respectfully requested.

With regard to amended claim 26 which is directed to a computer system for providing parallel execution of computing tasks in a heterogeneous computing environment, it comprises, *inter alia*, means for monitoring execution of the mobile agents at the computing hosts, means for detecting over-utilization of one of the computing hosts and for issuing a warning when one of the computing hosts is over-utilized and means for automatically transferring execution code as well as stack trace and state information about each of the mobile agents responsive to detected over-utilization of one of the computing hosts as claimed. Because Jagannathan does not operate in this manner, neither it nor the other references of record applied singularly or in combination teach or suggest this claim. Reconsideration and allowance of the claim is therefore respectfully requested.

With regard to amended claim 34 which is directed to a method for providing realistic thread migration, it comprises, *inter alia*, the step of detecting an indication to migrate said mobile agent thread and in response to said indication stopping execution of said mobile agent thread, automatically transferring the execution code where said mobile agent thread from a central server to a third computing host and automatically transferring said stack trace and state information about said mobile agent thread to said third computing host. Because neither Jagannathan or any of the other references of record operate in this way, they fail to teach or suggest this claim. Reconsideration and allowance of this claim is also respectfully requested.

In regard to amended claim 40 which claims an agent collaboration environment which comprises a plurality of mobile agents as well as an agent debugger for storing stack trace and state information about each of the plurality of mobile agents at a computing host where none of the plurality of mobile agents are executing, neither Reiffin nor Jagannathan, applied singularly or in combination with any of the other references of record, teach or suggest an agent debugger as claimed. Jagannathan maintains stack trace and state information at the very mobile agent that is executing.

Moreover, neither Jagannathan nor the other references of record teach or suggest a conference room for providing a virtual workspace for mobile agents or a registration subsystem for selectively assigning the plurality of mobile agents to the conference room wherein each of the plurality of agents can share data information in results of computations in the conference room. With regard to this operation, the examiner identifies column 18, lines 39-51 as teaching a virtual workspace for mobile agents, but a careful review of this text indicates such not to be the case. The identified text merely describes migrating two subagents A1 and A2 to a place where these subagents are merged into a subagent A3. It is submitted that this is not the same type of operation that is claimed and in fact in applicant's environment the plurality of agents maintain their identity but can share information and results of computations in the conference room.

Amended claim 42 is directed to a method for dynamically constructing and executing the computer application in a heterogeneous computing environment using mobile agents comprising, *inter alia*, the step of monitoring execution of each of the plurality of mobile agents and storing stack trace and state information about execution of the mobile agents at a computing host where none of the mobile agents are executing. For the reasons that were advanced with regard to claim 1, it is believed that this claim is also in condition for allowance and reconsideration and allowance of it is respectfully requested.

The dependent claims necessarily incorporate the subject matter of the claims from which they depend and in addition recite other features or functionality not found in those claims and are therefore also allowable. Reconsideration and allowance of all dependent claims is therefore respectfully requested.

For the foregoing reasons, the application should be passed to issue without delay. If issues remain that need to be addressed, the examiner is requested to contact the undersigned at the examiner's early convenience.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By 

Roger D. Greer

Registration No. 26,174

August 19, 2005

300 South Wacker Drive, Suite 2500  
Chicago, Illinois 60606  
(312) 360-0080  
Customer No. 24978